

HPCC Year One and Two Progress Report

" A Scientific Web-based Application for Global Tropical Cyclone Monitoring "

Hurricane Research Division NOAA/AOML, Miami, FL

The following is a progress report for the first two years of a three-year project entitled "A Scientific Web-based Application for Global Tropical Cyclone Monitoring".

Milestones and Deliverables (From Year 1 & 2 proposal)

05-31-2001

Producing a web-based prototype of the H*WIND system involves two parts:

Evaluation of Rapid Application Development tools such as Web Objects and the Oracle JDeveloper through the development of a baseline prototype of a small application to query the atmospheric observation database.

Develop Java components that encapsulate the query mechanisms to manipulate the atmospheric observation database. Specifically here, we will investigate distributed component technologies such as JavaBeans, EJB, and CORBA server objects that may reside in the Oracle8i database (EJB and CORBA objects only) or anywhere over the AOML Intranet. While developing the Java components, we will investigate the performance gain or loss in using JDBC and Oracle SQLJ implementations.

12-1-2001

Evaluation of XML, Java servlets and JavaServer Pages (JSPs) for dynamic content generation. Being Java-based Servlets and JSP programs can exploit the full power of the portability, safety, and extensibility of the object oriented Java platform.

07-1-2002

Develop EJB components to encapsulate the query mechanisms to manipulate the HURDAT and H*WIND databases. Investigate the performance gain or loss in using JDBC and Oracle SQLJ implementations.

12-1-2002

Develop Java servlets, JavaServer Pages (JSPs) and JavaBeans to communicate with the middle tier HURDAT and H*WIND EJBs and at the same time, to manipulate the data for the presentation layer. Improve front-end client through use of Java applets and possibly XML.

Progress by Milestone

A Web-based prototype of the H*WIND system. Users will be able to display tropical cyclone observations worldwide and interact with the data.

As of May 31, 2002, a Web-based prototype of the H*WIND system is available on <http://storm.aoml.noaa.gov/Demos.html>. This prototype allows users to retrieve, via the Web, storm track fixes and plot them on the fly. Users can request storm tracks per year and/or per basin (Atlantic, Central Pacific, East Pacific, West Pacific, South Pacific, and Indian). Note that, currently, our H*WIND database is running on the most updated version of the Oracle database software, Oracle9i Release 9.2.0.1.

Evaluated Oracle JDeveloper Rapid Application Development tool.

We evaluated the Oracle JDeveloper tool. We use the tool to develop database tier components using JDBC and/or SQLJ API. JDBC API (Java DataBase Connectivity Application Programming Interface) allows Java programs to access SQL databases in Java. SQLJ is an extension of JDBC that accomplishes the same tasks as JDBC but at a higher level.

Developed five Java components to encapsulate the query mechanism for manipulating HURDAT storm tracks. This process investigated the distributed component technologies JavaBeans, EJB, and CORBA server.

The Web-based prototype uses JDBC/SQLJ components to access the HURDAT storm tracks. We also investigated technologies such as JavaBeans, Enterprise JavaBeans, and CORBA. A JavaBean component is just a Java class that meets the JavaBean API requirements. The JavaBeans technology was developed to enable the programmers to rapidly build applications by assembling objects and test them during design time, thus making reuse of the software more productive. Enterprise JavaBeans (EJB) architecture specifies how communication among components maps into underlying communication protocols. EJBs are components that live and operate on an application server. We also investigated CORBA (Common Object Request Broker Architecture) an open standard for working with distributed objects.

We Developed EJB components to encapsulate the query mechanisms to manipulate the HURDAT and H*WIND databases.

We upgraded the prototype capabilities allowing the application to access the database via Enterprise JavaBeans and SQLJ components. Why EJB and SQLJ? Per our investigation and heavy testing of our components, we noticed that data ingestion via SQLJ was much faster than ingestion via EJBs.

Develop Java servlets, JavaServer Pages (JSPs) and JavaBeans to communicate with the middle tier HURDAT and H*WIND EJBs and at the same time, to manipulate the data for the presentation layer.

We developed Java Servlets, JavaServer Pages (JSPs), and JavaBeans to communicate the HURDAT database. These Java programs use the EJBs components to access the database.